

## REMARKS

Claims 1-57 were pending in the application. Upon entry of this Response, claims 2-20, 28-33 and 35-41 will be presented for further consideration, claims 1, 21-27, 34 and 42-57 having been canceled in this paper.

Minor informalities in the specification have been corrected as indicated above.

### Claim Objections

Claim 18 is objected to because of an informality therein. To overcome this objection, claim 18 has been amended to recite “a translator for preparing commands”.

### Claim Rejections under 35 USC § 112

Claims 2-20 are rejected under 35 U.S.C. 112, second paragraph, as being indefinite. It is noted that claim 8 has been rewritten in independent form and replaces former claim 1 as the only independent claim in this group of claims. In rewriting claim 8, the limitation of claim 1 which formerly recited “a command having an attribute” has been presented instead as “a command having input attributes and output attributes”, thereby overcoming the Examiner’s rejection on this point.

The Examiner also questioned the use of the phrase “performing a view” with respect to claim 7. This phrase has been deleted from claim 8. (It is noted that this phrase does not appear in claim 7.)

In view of the foregoing, it is requested that the rejection of claims 2-20 under § 112, second paragraph, be reconsidered and withdrawn.

### Claim Rejections under 35 USC § 102

Claims 33 and 35 are rejected under 35 U.S.C. 102(e) as being anticipated by Limondin et al. (U.S. Patent No. 6,226,783, hereinafter “Limondin”).

Claim 33 is directed to a “computer-readable medium containing a data structure defining a command-based program”. The data structure includes, for commands of the program, “a descriptor for the command, the descriptor identifying an object class, the object class defining input and output attributes and having a perform method”. Claim 33 further recites “whereby the program is executed by instantiating objects corresponding to the object classes of the commands, setting input attribute values of objects based on output attribute values of objects, and performing the perform method of the objects”. In addition, as now amended, claim 33 recites that “the commands are organized into interactions, one of the interactions including a first object class to retrieve a current application context of the application, a second object class to begin a transaction for an asset catalog, a third object class to identify a value of attributes of an asset to be added to the asset catalog, a fourth object class to store an entry identified by a created object in the asset catalog, and a fifth object class to indicate that the transaction is ending”.

Applicants wish to particularly point out that claim 33 has now been amended to include the limitation of former claim 34 and to further recite that one of the interactions includes a first object class to retrieve a current application context of the application, a second object class to begin a transaction for an asset catalog, a third object class to identify a value of attributes of an asset to be added to the asset catalog, a fourth object class to store an entry identified by a created object in the asset catalog, and a fifth object class to indicate that the transaction is ending. Support for the latter limitations is found at paragraph [0025] of the specification.

It is noted that neither the Limondin reference nor the other prior art of record teaches or suggests an interaction including the five object classes now specifically recited in claim 33. It is therefore felt that claim 33, as now amended, is allowable over the prior art of record. Claim 35, being dependent on claim 33, is believed allowable on the same basis as claim 33.

#### **Claim Rejections under 35 USC § 103(a)**

Claims 2-5, 15, 16, 19 and 20 are rejected under 35 U.S. C. 103(a) as being unpatentable over “Sams Teach Yourself Visual Basic 6 in 21 Days” by Greg Perry, Sams, 1998 (hereinafter Perry) in view of Silberbauer et al. Claims 8-11 and 14 are rejected as being unpatentable over

Perry in view of Silberbauer and further in view of Laskoski. As noted above, claim 8 has been rewritten in independent form and other claims have been amended so that all of claims 2-7 and 9-20 are directly or indirectly dependent on claim 8.

Claim 8 is directed to a “method in a computer system for executing a program”, including “providing a plurality of definitions of interactions of the program, an interaction having one or more command definitions defining commands and a view definition, a command having input attributes and output attributes and a behavior, each interaction being associated with a request”. The method of claim 8 further includes “receiving a request”, “identifying the interaction associated with the received request” and for each command of the identified interaction, “preparing the command by setting values of input attributes of the command based on attribute values stored in an attribute store”, “performing the behavior of the command with the input attributes” and “extracting attribute values of the command by retrieving values of the output attributes of the command and storing the retrieved values of the attributes in the attribute store”. The method of claim 8 also includes “providing a response generated by the view associated with the identified interaction”. Claim 8 further recites that “the view definition defines a view with attributes and a behavior”. In addition, claim 8 recites “setting values of input attributes of the view based on attribute values stored in an attribute store” and “performing the behavior of the view with the set attributes”.

As noted above, in rejecting claim 8, the Examiner proposed a combination of the Laskoski reference with the Perry and Silberbauer references. In particular, the Examiner relied upon Laskoski to supply the limitations relating to a view definition that defines a view with attributes and a behavior and the steps of setting values of input attributes of the view based on attribute values stored in an attribute store and performing the behavior of the view with the set attributes. Applicants respectfully traverse the Examiner’s proposal to combine Laskoski with Perry and Silberbauer.

The Perry reference discloses a programming language for creating programs which utilize a graphical user interface. By using the programming language of Perry, a programmer may create forms that have command buttons. The command buttons may be clicked by a user of the program to input commands.

Silberbauer discloses a system in which a work station may call functions from a host computer. Input parameter values for a function invoked on the host may be stored in and retrieved from a Common Data Format (CDF) data stream. Output parameters for the function are placed into the CDF data stream.

By contrast, the disclosure of the Laskoski reference has nothing to do with creating an interface to allow user command input or calling functions from a host computer. Rather, the main point of Laskoski has to do with an improvement in a program which graphically illustrates calls between functions to aid in analyzing a computer program. A person of ordinary skill in the art who was concerned with providing a program having a visual interface for command entry and/or program functions that are callable from a host computer would have no reason to incorporate in such programs features of a graphical program analysis utility such as that disclosed in Laskoski. Such a utility would be of value only for a programming professional who was interested in the relationships among functions in a program, and not for the ordinary users of the types of programs described in the Perry and Silberbauer references.

Furthermore, the motivation proposed by the Examiner for combining Laskoski with Perry and Silberbauer is not supported by those references. The Examiner asserts (at the bottom of page 8 and the top of page 9 of the pending Office Action) that Laskoski "allows output of a command to be displayed on a computer screen so that a user can view output data". It is respectfully pointed out that this is not a correct characterization of Laskoski's disclosure. Laskoski is concerned with graphically displaying calls between functions of a computer program that is analyzed by a utility, and is not concerned with allowing a user to "view output data". Thus the Examiner has not provided a sound rationale for the proposed combination of Laskoski with Perry and Silberbauer.

It is therefore respectfully submitted that the teachings of Laskoski cannot properly be combined with those of Perry and Silberbauer, and that the feature of Laskoski relied upon by the Examiner cannot properly be imported into the method derived by combining Perry and Silberbauer, since there is lacking any suggestion in the prior art that such a combination be made. Thus, considering the references as a whole, the method recited in claim 8 is not obvious. It is therefore requested that the rejection of claim 8 be reconsidered and withdrawn.

Claims 2-7 and 9-20 are now dependent on claim 8, and are believed to be patentable on the same basis as claim 8.

In addition, claim 17 recites the limitation that “a command definition indicates whether the behavior of the command should be performed when an error is detected while performing the behavior of the commands of the interaction”. The Examiner’s rejection of claim 17 relied upon an asserted combination of the Eisenberg reference with the Perry and Silberbauer references. In particular, the Examiner conceded that Perry and Silberbauer fail to disclose a command definition which indicates whether the behavior of the command should be performed when an error is detected when performing the behavior of the command. However, the Examiner then relied on the teaching of Eisenberg of indicating upon an error in a program whether the program should continue or halt. The Examiner considered it obvious to incorporate this feature from Eisenberg in a program formed by combining Perry and Silberbauer.

Applicants respectfully traverse this rejection. Eisenberg is concerned with programming for a safety critical application such as a medical device (e.g., an MRI imaging system). There is nothing in the Perry and Silberbauer references that is concerned with programming for safety critical applications, and there accordingly is no reason why one of ordinary skill who was working with the types of programs described in Perry and Silberbauer would consult a reference such as Eisenberg. There is no suggestion in the prior art to support the combination of references made by the Examiner with respect to claim 17, and it is accordingly submitted that the proposed combination was improper.

Moreover, Eisenberg fails to teach the specific mechanism recited in claim 17 for determining whether to perform a command. That is, Eisenberg fails to teach a command definition to indicate whether the behavior of the command should be performed in case of error. Rather, Eisenberg discloses a set of safety critical rules by which unsafe operation may be prevented. This is not the same as the command definition recited in claim 17.

For these reasons, claim 17 is believed to present grounds of patentability that are in addition to the basis for patentability of its parent claim 8.

Claim 28 is rejected as being unpatentable over Travostino et al. U.S. Patent No. 6,564,325 (“Travostino”) in view of Shapiro et al. U.S. Patent No. 5,257,363 (“Shapiro”).

Claim 28 is directed to a “method in a computer system for generating source code for a program” including “receiving a list of names of functions to be invoked by the program” and, for each of the functions to be invoked of the program, “identifying names of formal parameters of the function” and “outputting an invocation of the function that includes names of actual parameters derived from the identified names of formal parameters whereby the output invocations of the functions form the source code for the program”.

In explaining the rejection of claim 28, the Examiner conceded that the Travostino reference fails to teach, for each function, identifying the names of formal parameters of the function and outputting an invocation of the function with names of actual parameters derived from the formal parameters. The Examiner then proposed to compensate for this defect of the Travostino reference by relying on the teaching of Shapiro in regard to supplying a list of actual parameters in place of formal parameters.

In this regard, applicants respectfully urge that the Shapiro reference fails to teach identifying names of formal parameters or outputting an invocation of a function that includes names of actual parameters, as specifically recited in claim 28. Rather, Shapiro only teaches (at column 20, lines 11-17) supplying the actual parameters for a subroutine declared with formal parameters. Nothing is said in Shapiro of the names of the formal or actual parameters.

Thus, the references fail to teach particular limitations recited in claim 28, and it is therefore respectfully requested that the rejection of claim 28 be reconsidered and withdrawn.

Claims 29-32 are directly or indirectly dependent on claim 28 and are submitted as patentable on the same basis as claim 28.

Claims 36-41 are directly or indirectly dependent on claim 33 and are submitted as patentable on the same basis as claim 33.

### Conclusion

Accordingly, Applicants respectfully request allowance of the pending claims. If any issues remain, or if the Examiner has any further suggestions for expediting allowance of the

present application, the Examiner is kindly invited to contact the undersigned via telephone at (203) 972-3460.

Respectfully submitted,



January 27, 2004

Date

---

Nathaniel Levin  
Registration No. 34,860  
Buckley, Maschoff & Talwalkar LLC  
Five Elm Street  
New Canaan, CT 06840  
(203) 972-3460